A HIGH THROUGHPUT ADAPTIVE DFE FOR HIPERLAN

R. Perry, D. R. Bull and A. Nix
Centre for Communications Research
University of Bristol, Bristol, BS8 1TR, UK.
email russ.perry@bristol.ac.uk, dave.bull@bristol.ac.uk,
Tel: +44 117 928 7740 Fax: +44 117 925 5265

Abstract

This paper describes two methods for increasing the throughput of an adaptive Decision Feedback Equaliser (DFE) using the LMS training algorithm. In the first method, a signed power-of-two number representation is used for the equaliser input data. Using this number representation, all multipliers can be replaced with barrel shifters and adders. In the second method, the Delayed Least Mean Square Algorithm (DLMS) is used to train the equaliser. A delay, equal to the feedforward filter length, is introduced in the filter coefficient update, which allows the DFE to be realised as the cascade of a series of modular sections.

1. INTRODUCTION

This paper discusses methods of improving the throughput rate of an adaptive equaliser. Such techniques are of current interest in the context of emerging high data rate wireless LANs such as HIPERLAN [1]. The HIPERLAN supports data rates of up to 23.5Mb/s which, even in indoor environments, can lead to very severe intersymbol interference (ISI). This necessitates the use of an adaptive equaliser in the receiver.

Numerous equaliser algorithms and architectures have been reported in the literature. Decision Feedback Equalisation is considered here, since the complexity of alternatives such as maximum likelihood sequence estimation is prohibitive for channel impulse response lengths greater than 5 symbols. A DFE can be realised, using either transversal filters, lattice filters or systolic arrays [6]. In [6] adaptive equalisers were considered for application to TDMA based systems, which in some cases, impose severe tracking requirements on the equaliser. However, in the case of HIPERLAN, reasonably stationary channel conditions can be assumed. The equaliser is trained using all, or part of, a 450-bit header packet and may then be fixed while the following data blocks (up to 49) are processed. The comparatively long training sequence allows low complexity (slow converging) algorithms to be used for equaliser training. For this reason the LMS algorithm or a variant is a natural choice. However, achieving a throughput of 23.5Mb/s is still problematic due to the sampling rate limitation associated with the coefficient update and the decision feedback loop, imposed by a conventional DFE structure.

Two methods for producing a reduced complexity high throughput rate DFE are described in this paper. The motivation for choosing a transversal filter is explained in section 2.1. The complexity of the modified DFE architectures and their convergence and output mean square error characteristics are discussed in sections 3 and 4 respectively.

2. ADAPTIVE TRANSVERSAL EQUALISER's

Two methods for increasing the throughput rate of a transversal filter based DFE are described in this section.

2.1. *Non-Uniform Number Representation*

The first method uses non-uniform quantisation (a signed-power-of-two (SPT) approximation) of the equaliser input data [3]. This quantisation is applied to the input data, as opposed to the filter coefficients, since the performance of the equaliser is largely unaffected by this approximation (see section 4) while facilitating significant complexity savings. In addition, the non-uniform quantisation of the input data is required only *once* per input sample. In contrast, for SPT filter coefficients, it is necessary to quantise the coefficients following each update. This introduces additional latency and complexity within the coefficient update loop. The standard LMS algorithm can be used in both cases without modification.

A representation of a discrete-time B-bit two's complement number $x(m)$, in the signed power-of-two space [3] is given by

$$x_N(m) = \sum_{r=1}^{N} s(r)2^{g(r)}, \quad s(r) = -1, 0, 1 \qquad (1)$$

where $g(r)$ is the power of the r^{th} power-of-two (POT) term and N is the number of POT terms used in the approximation. If $x(m)$ is an integer, then for $N = \lceil B/2 \rceil$, all integers in the range $-2^{B-1} \dots 2^{B-1} - 1$ can be exactly represented. However, for $N < \lceil B/2 \rceil$, not all integer values that $x(m)$ may take, can be represented by $x_N(m)$. Hereafter the term, N-SPT, will be used to denote an approximation (in some cases an exact representation) of a two's complement integer using N POT terms, each taking either positive or negative sign.

0-7803-3073-0/96/$5.00 ©1996 IEEE

The area and or latency of a multiplier can be significantly reduced by using restricted-number representations for either the multiplier or multiplicand, i.e. using coefficients with a limitation on the number of non-zero digits. The multiplier can then be replaced with shift and addition elements. Using a 2-SPT representation of the input data, as described above, allows the multipliers in both the transversal filter and coefficient update modules (for the LMS algorithm) to be replaced with a pair of barrel shifters and a single adder.

The transversal filter based DFE operates directly on the input data samples (as opposed to the backward residuals in the case of a lattice structure [5]) and is therefore the natural choice for exploitation of the SPT data representation.

2.2. *Pipelined DLMS DFE*

Adaptive transversal filters suffer from an inherent sampling rate limitation for a given speed of hardware. This is due to the feedback of the residual error necessary to adapt the filter coefficients, i.e. the whole residual error calculation must be completed before the coefficient update can be performed.

The throughput bottleneck described above can be overcome using the DLMS algorithm [4] This is an approximation of the LMS algorithm offering a modular, high throughput filter structure with clock rate limited only by the delay in a single processing module. The modified structure also operates directly on the input data stream, again facilitating savings from using the SPT number representation. It is shown in section 4 that the degradation in performance when using the DLMS algorithm is not significant.

Previously, the DLMS algorithm has been employed to allow pipelining of the LMS algorithm for a linear structure [4]. This method is extended here to allow a modular high throughput DFE structure to be developed. The DLMS algorithm is given by equations (2) and (3) [4, 9]

$$\mathbf{W}(n) = \mathbf{W}(n-1) + \beta e^*(n-D)\mathbf{X}(n-D) \tag{2}$$

$$e(n) = d(n) - \mathbf{W}^H(n-1)\mathbf{X}(n) \tag{3}$$

The $\mathbf{W}(t)$ and $\mathbf{X}(t)$ vectors are first partitioned into the feedforward and feedback sections respectively; (2) is then rewritten

$$\begin{aligned}\left[\mathbf{W}_f(n), \mathbf{W}_b(n)\right] = \left[\mathbf{W}_f(n-1), \mathbf{W}_b(n-1)\right] + \\ \beta e^*(n-D)\left[\mathbf{X}_f(n-D), \mathbf{X}_b(n-D)\right]\end{aligned} \tag{4}$$

where

$$\mathbf{X}_f(n) = \left[x_f(n) \;\; x_f(n-1) \;\; \ldots \;\; x_f(n-L+2) \;\; x_f(n-L+1)\right]$$
$$\mathbf{X}_b(n) = \left[x_b(n) \;\; x_b(n-1) \;\; \ldots \;\; x_b(n-L+2) \;\; x_b(n-L+1)\right]$$

which is the vector of previously detected symbols i.e.,

$$\mathbf{X}_b(n) = \left[d(n-1) \;\; d(n-2) \;\; .. \;\; d(n-L+1) \;\; d(n-L)\right]$$

In a manner similar to [4], an output vector is defined as

$$\mathbf{Y}(n) = \left[y_0(n) \;\; y_1(n-1) \;\; \ldots \;\; y_{L-2}(n-L+2) \;\; y_{L-1}(n-L+1)\right]$$

where

$$y_i(n-i) = \sum_{k=0}^{i} x_f(n-i-k)w_f^{k*}(n-i-1) + y_b(n-i) \tag{5}$$

and $y_b(n-i)$ is the contribution to the current equaliser output from the feedback filter. Rewriting (5) gives

$$\begin{aligned}y_i(n-i) = \sum_{k=0}^{i-1} x_f(n-i-k)w_f^{k*}(n-i-1) + \\ x_f(n-2i)w_f^{i*}(n-i-1) + y_b(n-i)\end{aligned}$$

$$y_i(n-i) = y_{i-1}(n-i) + x_f(n-2i)w_f^{i*}(n-i-1) + y_b(n-i) \tag{6}$$

Initially we define $y_b(n-i)$ as

$$y_b(n-i) = \sum_{k=0}^{L-1} x_b(n-i-k)w_b^{k*}(n-1-k) \tag{7a}$$

This can be interpreted as a transposed direct-form transversal filter. The delay k, in the coefficient terms in (7a) is due to the delay in the output signal propagating along the filter structure. This is not a strict realisation of the DLMS algorithm. However, by inserting delays in the filter coefficient terms in (7a), a transposed filter structure implementing the DLMS algorithm is obtained i.e. the kth coefficient used in (7a) should be delayed by L-1-k iterations. In this case $y_b(n-i)$ is given by

$$y_b(n-i) = \sum_{k=0}^{L-1} w_b^{k*}(n-k-1-(L-1-k))x_b(n-i-k)$$

$$y_b(n-i) = \sum_{k=0}^{L-1} w_b^{k*}(n-L)x_b(n-i-k) \tag{7b}$$

The transformed block diagram for a (3,3) DFE using the DLMS algorithm is shown in figure 1; and consists of three identical processing modules (PMs). The latency in the output is 2L-1 sample periods. This is the time required for all the feedforward filter stages to *fill* and for the estimate of the desired response to propagate along the filter structure.

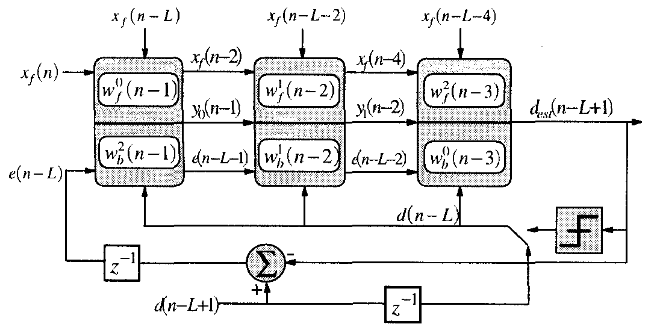

Figure 1: Transformed (3,3) DFE Structure

It should be noted that the input to the feedforward filter enters from the left whilst the previous decision is input to *all* the feedback filter sections simultaneously. Note also that the index for the feedforward filter coefficients increases left to right, but for the feedback filter coefficients, it decreases left to right.

The weight update for $w_f^i(n-i-1)$ required by (5) is obtained from (4) as

$$w_f^i(n-i) = w_f^i(n-i-1) + \beta e^*(n-L-i)x_f(n-L-2i) \tag{8}$$

For the update of $w_b^i(n-1)$ there are two forms corresponding to equations (7a) and (7b). For (7a) the weight update is

$$w_b^i(n) = w_b^i(n-1) + \beta e^*(n-L)x_b(n-L-i) \quad \text{(9a)}$$

For (7b) the weight update is given by

$$w_b^i(n-(L-1-i)) = w_b^i(n-L+i) + \beta e^*(n-2L+1+i)x_b(n-2L+1) \quad \text{(9b)}$$

In both (9a) and (9b), global communication is required; in (9a) the same error term is fed back to all the coefficient update sections, whereas in (9b) the same data symbol is fed back. The form (9b) is attractive because the feedback data is only a complex number of the form $\pm 1 \pm j$. In addition, because of the reversed order of the feedback filter coefficients, the error term in (8) is the same as that required in (9b) and therefore this reduces the communication costs considerably. An individual PM for the DLMS DFE structure is shown in figure 2 using the update (9b). The throughout of a DFE, is limited now, by the time to perform a multiply shift and add (M6-M5-A3). An N-SPT approximation for the input data can also be used to reduce the complexity of the proposed filtering structure, as described for the LMS algorithm.

The new DFE structure has the additional advantage that different representations for the feedback and feedforward input data do not destroy the regularity of the structure. This is in contrast with the conventional LMS algorithm, where different feedforward and feedback filter structures would be required for the different number representations of the input data .

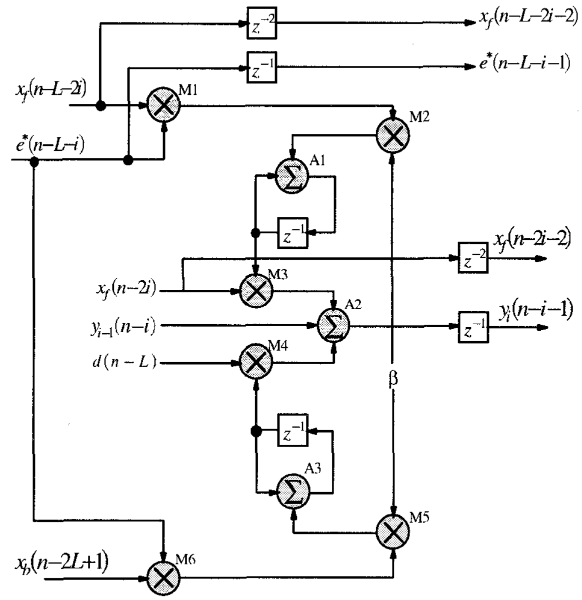

Figure 2: Individual PM for the DLMS Algorithm

3. COMPLEXITY COMPARISON

In order to determine the potential complexity (area) savings from using 2-SPT feedforward input data, the gate counts required to implement LMS and DLMS based DFEs using both two's complement and 2-SPT input data have been estimated (figure 3). It is assumed in all cases, that the step size parameter is selected as a POT term to eliminate one full multiplier and that single POT terms are used for the feedback data. The gate counts used for each type of logic gate are based on commercial products [7,8]. Each *gate-equivalent* is a structure from which a 2-input NAND gate, or a 2-input NOR gate, can be constructed. A Baugh-Wooley parallel multiplier (for two's complement data) and a Barrel Shifter Multiplier (for SPT Data) are assumed. The filter length of the feedforward and feedback filter L, is fixed at L=8, as this is anticipated to be the longest filter required for a HIPERLAN equaliser.

It can be seen from figure 3 that the use of the SPT coded data reduces the gate count by 50% for an input wordlength of 8 bits, compared to a two's complement representation. The complexity of the DLMS algorithm differs from LMS algorithm only by the additional pipelining latches.

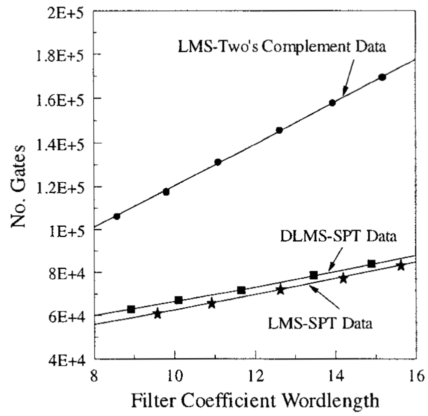

Figure 3: Complexity of LMS & DLMS DFEs

4. CONVERGENCE & RESIDUAL MSE

The effect of the non-uniform approximation of the input signal on the equaliser's performance is considered here. For comparison, a stationary channel characteristic leading to an eigenvalue spread of 46 [9] is used to distort a QPSK signal. Additive noise is added ($E_b/N_0 = 20\text{dB}$) and the signal is root-raised cosine filtered. The convergence of a (3,3) DFE using the DLMS algorithm and 2-SPT input data (approximation obtained from a linearly quantised 8-bit input data stream) is compared with the LMS algorithm using the original 8-bit input data in figure 4. For clarity, only a small number of points have been plotted for the LMS algorithm. It can be seen that the effect of the algorithm approximation and non-uniform quantisation of the input data has had no significant effect on the convergence behaviour of the equaliser. The step size was chosen to be the same in both cases.

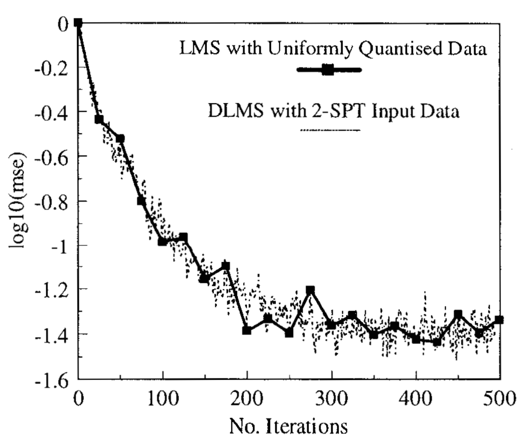

Figure 4: Convergence of LMS and DLMS Algs. (δ= 0.0625)

5. CONCLUSIONS

This paper has discussed methods to reduce the complexity and increase the throughput of adaptive transversal DFEs for applications such as HIPERLAN. It was shown that the use of a 2-SPT approximation of the input data allows a reduction in gate count of up to 50% for filter coefficients of 16-bits and a filter length of 8. A new modular structure for implementing a pipelined DFE using the DLMS algorithm was also described. This modified structure resulted in a throughput rate determined by a single multiplier, barrel shifter and adder. Using non-uniform quantisation of the input data in conjunction with this structure allows the throughput rate to be improved still further.

ACKNOWLEDGEMENTS

The authors wish to express their gratitude to the members of the Centre for Communications Research, University of Bristol. We also wish to thank EPSRC and Hewlett Packard Laboratories for their support of this work.

REFERENCES

[1] Etsi Radio Equipment and Systems "HIgh PErformance Radio Local Area Network (HIPERLAN)," Functional Specification Version 1.1 (Draft), Jan. 1995.

[2] Y. C. Lim, J. B. Evans, B. Liu, "Decomposition of binary integer into Signed Power-Of-Two Terms," IEEE Trans on Circuits Systs., vol. CAS-38, pp. 667-672, June 1991.

[3] D. Li, Y. C. Lim, "Multiplierless Realization of Adaptive Filters by Nonuniform Quantization of Input Signal," ISCAS 1994, vol. 2, pp. 457-460.

[4] M. D. Meyer, D. P. Agrawal "A High Sampling Rate Delayed LMS Filter Architecture," *IEEE Trans on Circs & Systs*, vol CAS-40 II, Nov 1993, pp 727-729.

[5] T. H.-Y. Meng, D. G. Messerschmidt "Arbitrarily High Sampling Rate Adaptive Filters," *IEEE Trans Acoustics, Speech and Sig. Proc.*, vol ASSP-35, no. 4, April 1987, pp 455-470.

[6] R. Perry, D. Bull, A. Nix "Algorithms for Flexible Equalisation in Wireless Communications," *IEEE Proc.* ISCAS April 1995, vol 3., pp. 1940-1943.

[7] LSI Logic 1.0-Micron Cell-Based Products Data Book LCB007 - Cell Based ASIC's Feb. 1991

[8] LSI Logic 1.0-Micron Array-Based Products Data Book LCA100K Compacted Array Plus™ Series LEA 100K Embedded Array Series, Sept 1991.

[9] S. Haykin "Adaptive Filter Theory," Prentice Hall 1991 2nd Ed.